• ¿Para qué sirve la variable **chunk**?

A la variable **chunk** se le asigna el valor de la variable global **CHUNKSIZE** y posteriormente se usara en la cláusula **Schedule** junto a la palabra **dynamic** que asigna una iteración de la estructura **for** posterior a cada hilo (Cuando el hilo finaliza se asignara la siguiente iteración no ejecutada). La variable **chunk** tiene un objetivo distinto que es el de dar el tamaño al bloque de datos de los hilos.

• Explique completamente el **pragma****“#pragma omp parallel shared(a,b,c,chunk) private(i)”**. ¿Por qué y para qué se usa **shared(a,b,c,chunk)** en este programa?

Mediante la clausula **parallel** indicamos que la estructura que viene a continuación tendrá una ejecución paralela, es decir, la siguiente estructura se verá afectada por el **pragma** y **parallel** puede tomara los núcleos de manera automática y paralelizara la siguiente estructura.

Mediante la clausula **shared** declaramos las variables citadas en esta como variables publicas que, serán compartidas en el espacio de disco público.

¿Por qué la variable i está etiquetada como **private** en el **pragma**?

Todas las variables son públicas por defecto pero, mediante la clausula **prívate,** declaramos las variables citadas en esta como variables privadas dejándolas para la ejecución de un único núcleo.

• ¿Para qué sirve **schedule**? ¿Qué otras posibilidades tiene?

Describe como las iteraciones del bucle son divididas entre los hilos del mismo grupo. La variable **chunk** tiene un objetivo distinto que es el de dar el tamaño al bloque de datos de los hilos. La palabra **dynamic** asigna una iteración de la estructura **for** posterior a cada hilo (Cuando el hilo finaliza se asignara la siguiente iteración no ejecutada). Además de la palabra **dynamic** también podemos encontrar otras posibilidades como:

**Static**: Las iteraciones del bucle son divididas en piezas del tamaño de **chunk** y estáticamente asignadas a los hilos. Las iteraciones serán divididas equitativamente entre los hilos si la variable **chunk** no es especificada.

**Guided**: Las iteraciones son dinámicamente asignadas a los bloques de hilos conforme los hilos las vayan pidiendo hasta que ningún bloque quede para ser asignado.

**Runtime**: El uso de **Schedule** viene determinado en la ejecución de la **variable OMP\_SCHEDULE.**

**Auto**: Sera el compilador el que automáticamente decidirá el **Schedule.**

**Ordered:** Especifica que las iteraciones del bucle deben ser ejecutadas de la misma forma que serian ejecutas en un programa en serie.

**Collapse:** Especifica cuantos bucles deben haber en un largo espacio de iteración y dividido conforme la clausula **Schedule** dicte.

Además de **nowait** también podemos emplear el uso de **wait.** La opción **wait** es usada para permitir la sincronización de los procesos y **nowait** tiene el uso contrario.